Running a Singularity Container Image on Pleiades

Frequent Updates: Support for Singularity is a new offering on NAS systems. The articles in this section are still under development and may be updated frequently as new usage information becomes available.
If you have an existing Singularity image that is ready to run, you should be able to port and run it on any platform where Singularity is installed. This article provides information to help you run a Singularity container image on Pleiades.

Note: A container is a runtime instance of an image. You can run multiple containers of the same image.

Unpacking the Image Tar File on Pleiades

If you transferred your Singularity container image to Pleiades as a compressed tar file, use the appropriate tool to uncompress and unpack the tar file. For example:

```
pfe% cd /nobackup/username/dir_containing_the_tar_file
pfe% tar -xzf your_image_name.tgz
```

Accessing Singularity on Pleiades

To find out what versions of Singularity are installed, run:

```
pfe% module avail singularity
singularity/3.x.y    singularity/3.x.z
```

To load the latest version (currently, version 3.9.8), run:

```
pfe% module load singularity
pfe% which singularity
/nasa/singularity/3.9.8/bin/singularity
```

To load a specific version, run:

```
pfe% module load singularity/3.x.z
pfe% which singularity
/nasa/singularity/3.x.z/bin/singularity
```

Running Your Container Image

Depending on what you want to do, you can use one of the following three commands to run the container image. If you include the **`--writable`** option when you run your container, you can also write files within the sandbox directory, provided you have the permissions to do so.

`shell`
: The **`singularity shell`** command allows you to spawn a new Bourne shell within your container and interact with it as though it were a small virtual machine. To exit the shell, enter **`exit`** at the Singularity prompt.

`exec`
: The **`singularity exec`** command allows you to execute a custom command within a container.

`run`
: Singularity containers may contain runscripts. These are user-defined scripts that define the actions a container should perform when it is run. The runscript can be triggered with

the **`singularity run`** command, or simply by calling the container as though it were an executable.

Running Your Container Image: Examples

For example, if you port the lolcow sandbox container to Pleiades, you can run it as described in the following examples.

Notes:

- With Singularity version 3.9.8, you cannot **`shell/exec/run`** the sandbox on your $HOME filesystem. Use it on your Lustre /nobackup filesystem, instead.
- You can also run with a Singularity Image file (**`*.sif`**) instead of a sandbox container. For example, replace the sandbox directory **`lolcow`** in Examples 1-3 with the **`.sif`** file named **`lolcow.sif`**.

Example 1: Using the shell Command

```
pfe% singularity shell lolcow
Singularity> fortune | cowsay | lolcat
 _________________________________________
< You enjoy the company of other people. >
 -----------------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||

Singularity> exit
```

Example 2: Using the exec Command

```
pfe% singularity exec lolcow which cowsay
/usr/games/cowsay
pfe% singularity exec lolcow cowsay hello
 _______
< hello >
 -------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
pfe%
```

Example 3: Using the run Command

```
pfe% singularity run lolcow
 ___________________________________
< It's all in the mind, ya know.  >
 -----------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
pfe%
```

Running Graphical Applications in the Container

If your container image includes some graphical applications (for example, xclock or xeyes), you will have to make sure that the DISPLAY environment variable is set inside the container in order to run them. For example,

```
pfe% singularity shell -w your_image_sandbox
Singularity> echo $DISPLAY
ii.jjj.kk.ll:76.0
Singularity> xclock
```

where *ii.jjj.kk.ll* represents the IP address of the specific PFE host. (Its actual value will be different for different hosts.)

Note: Using the **-e** option of the **singularity** command will clean the environment, including the setting of DISPLAY, before running the container. If the use of **-e** is necessary for running your container, you can use the **--env** option to pass the DISPLAY variable to the container to run graphical applications.

```
pfe% singularity shell -e --env DISPLAY=$DISPLAY -w your_image_sandbox
```

Note: If you encounter the following warning message, **cd** to *your_image_sandbox* and use the **mkdir** command to create **home*X*** before you retry the commands above. (The *X* in **home*X*** represents the actual value (*X*=1-7) of your $HOME directory on Pleiades.)

```
WARNING: By using --writable, Singularity can't create /homeX destination
automatically without overlay or underlay
```

Setting Up Environment Variables

You can modify environment variables, such as $PATH, within your container. For example:

```
pfe% singularity shell lolcow
Singularity> echo $PATH
/usr/games:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
Singularity> export PATH=$PATH:/homeX/username/bin
Singularity> echo $PATH
/usr/games:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/homeX/username/bin
```

Accessing Files on Pleiades

You can access files on Pleiades from within the container. By default, Singularity bind-mounts $HOME and /tmp into your container at runtime. With recent versions of Singularity, such as version 3.9.8, $PWD inside the container defaults to your Pleiades $HOMEâ *not* to the $PWD on your Pleiades host. For example:

```
pfe% cd /nobackup/username/dir_containing_lolcow_image
pfe% pwd
/nobackup/username/dir_containing_lolcow_image
pfe% singularity shell lolcow
Singularity> pwd
/homeX/username
Singularity> cd /
Singularity> ls
bin   core  environment  home   lib    media  opt   root  sbin         srv  tmp  var
boot  dev   etc          homeX  lib64  mnt    proc  run   singularity  sys  usr
```

You can specify additional directories to bind-mount into your container with the **--bind** option. For example, if you want to be able to access files and directories under your **/nobackupp*X*/$USER** directory, you can do:

```
pfe% singularity exec --bind /nobackupp*X*/*username*:/mnt lolcow ls /mnt
or
pfe% singularity exec --bind /nobackup/*username*:/nobackup/*username* lolcow ls /nobackup/*username*
```

For read-only filesystems such as **/nasa**, optionally add **:ro** to the command as follows:

```
pfe% singularity exec --bind /nasa:/nasa:ro lolcow ls /nasa
```

You can bind-mount multiple filesystems. For example:

```
pfe% singularity shell --bind /nasa:/nasa:ro --bind /nobackupp*X*/*username*:/mnt lolcow
Singularity> cd /nasa
Singularity> pwd
/nasa
Singularity> cd /mnt
Singularity> pwd
/mnt
Singularity> exit
pfe%
```

Article ID: 638
Last updated: 16 Jun, 2022
Revision: 52
Cloud Computing -> Containers -> Singularity -> Running a Singularity Container Image on Pleiades
https://www.nas.nasa.gov/hecc/support/kb/entry/638/